

NASA Technical Memorandum 104735

1N-62

15141

P. 23

Network Interface Unit Design Options Performance Analysis

Frank W. Miller

June 1991

(NASA-TM-104735) NETWORK INTERFACE UNIT
DESIGN OPTIONS PERFORMANCE ANALYSIS (NASA)
23 p CSCL 09B

N91-24792

Unclas

G3/62 0015141



National Aeronautics and
Space Administration



NASA Technical Memorandum 104735

Network Interface Unit Design Options Performance
Analysis

Frank W. Miller
Lyndon B. Johnson Space Center
Houston, Texas

National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas

June 1991

TABLE OF CONTENTS

| | |
|-------------------------------------------------|----|
| List of Figures | 3 |
| 1 Introduction | 4 |
| 2 Architectural Options | 5 |
| 3 Analysis | 6 |
| 3.1 Latency | 8 |
| 3.1.1 Baseline Option | 8 |
| 3.1.2 Dual-ported Auxiliary Memory Option | 8 |
| 3.1.3 Common Bus Option | 9 |
| 3.1.4 Calculations and Results | 9 |
| 3.2 Throughput | 12 |
| 3.2.1 Baseline Option | 13 |
| 3.2.2 Dual-ported Auxiliary Memory Option | 14 |
| 3.2.3 Common Bus Option | 15 |
| 3.2.4 Calculations and Results | 15 |
| 4 Implementation Impacts | 17 |
| 4.1 Physical Implementations | 17 |
| 4.2 Communications and Tracking (C&T) Interface | 18 |
| 5 Summary | 20 |
| References | 21 |

LIST OF FIGURES

| | | |
|---|-----------------------------------------------|----|
| 1 | NIU Architecture Options | 6 |
| 2 | NIU Packet Latencies | 11 |
| 3 | NIU Throughput (packets/sec) | 16 |
| 4 | NIU Throughput (bits/sec) | 16 |
| 5 | C&T Interface to the NIU Architecture Options | 19 |

1 INTRODUCTION

This paper presents an analysis of three architecture options for the design of the Space Station *Freedom* Data Management System (DMS) Network Interface Unit (NIU). The NIU provides the interface from the Fiber Distributed Data Interface (FDDI) core network to the DMS processing elements.

The current requirements for the performance of the NIU are stated as follows. First, there is a general throughput requirement which is stated in the document, JSC 31000 [9]:

Sect. 3.3.2.7.1.1.3: NETWORK INTERFACE THROUGHPUT [PMC] [AC]

The DMS Network Interface Function shall provide an aggregate user data throughput (excluding all communications systems overhead) of 10 Mbps. The Network Interface Function shall be able to send this traffic to and receive it from any node on the DMS network, including the node(s) to the C&T system.

Second, there are maximum message latency requirements levied on communications which are correlated to message priorities. These requirements are also stated in JSC 31000 as

Sect. 3.3.2.7.1.1.6: LATENCY REQUIREMENTS [PMC] [AC]

The Network Interface Function shall satisfy the following latency requirements between nodes operating at engineered capacity (i.e. the total throughput stated above).

| <u>Traffic Priority</u> | <u>Mean Delay (ms)</u> | <u>95% (ms)</u> |
|-------------------------|------------------------|-----------------|
| Emergency | < 20 | < 25 |
| Expedited | < 20 | < 25 |
| Normal | < 50 | < 80 |
| Background | < 80 | < 150 |

Determination has been made by both NASA/JSC and the Work Package 2 (WP-2) contractor that the current baseline NIU design does not meet the specified *throughput* requirement. The subsequent analysis will support this statement. As a result, the WP-2 contractor has proposed an alternate design which makes substantial modifications to the NIU hardware. In addition, the Flight Data Systems Division (FDSD) has proposed a second alternative.

This paper addresses the stated requirements as they apply to the three hardware architecture options. Section 2 describes these options in detail. Section 3 provides the analysis of these options in terms of the stated requirements. Section 4 discusses the

implications of adopting one of the two implementation changes. Finally, Section 5 provides a summary.

2 ARCHITECTURAL OPTIONS

Figure 1 shows the three potential designs for the Space Station *Freedom* DMS NIU hardware architecture considered in this analysis. The first option represents the *current design baseline*. The second option is termed the *dual-ported auxiliary memory* option and the final option is referred to as the *common bus*.

Each option in Figure 1 illustrates the three cards which would be affected. The first card involved is the Network Interface Adapter (NIA). The purpose of this card is to provide a buffer for data which is either received from or is to be transmitted to the FDDI network medium. The second card is the Network Interface Unit Embedded Data Processor (NIU EDP). Its intended purpose is to provide a hardware platform for the Network Operating System (NOS) software. The intent of these two boards is to offload network communications processing from the third board, the Applications Embedded Data Processor (App EDP). The provision of this additional capability frees the computational requirements necessary to accomplish network communications which would be needed in the App EDP and makes it available for other applications software.

Also included in this diagram are illustrations of the backplane busses which are used in each option. In the first two options, IBM's Microchannel bus is used to effect data transfers between the NIA and the NIU EDP. A Multibus II backplane is then used to accomplish transfers between the NIU EDP and the App EDP. In the Common Bus option, a single backplane connects all three boards and is used to effect both of the specified data transfers.

NOTE: In this analysis, the Multibus II has been used to provide parameters for the analysis of a common bus architecture's performance. Use of the Microchannel bus in Bus Master mode in place of the Multibus II as the common backplane would be expected to produce similar performance results.

The final elements of this diagram are the numbered data transfer paths. These paths are used in the subsequent analysis and assume data transfers originate on the FDDI network and are destined for the App EDP. Equivalent transfers which originate in the App EDP and are bound for the FDDI network will use equivalent paths in the first two options. Equivalent performance for data passing in either direction is expected in these two approaches. In the third option, data originating in the App EDP which is destined for the FDDI media would have different end points for paths 2 and 4. Transfers which originate in the NIA buffer would instead originate in the App EDP. Path 2 would end in the NIU EDP memory and path 4 would end in the NIA buffer. Despite this reversal, equivalent performance for data passing in either direction is also expected in this approach.

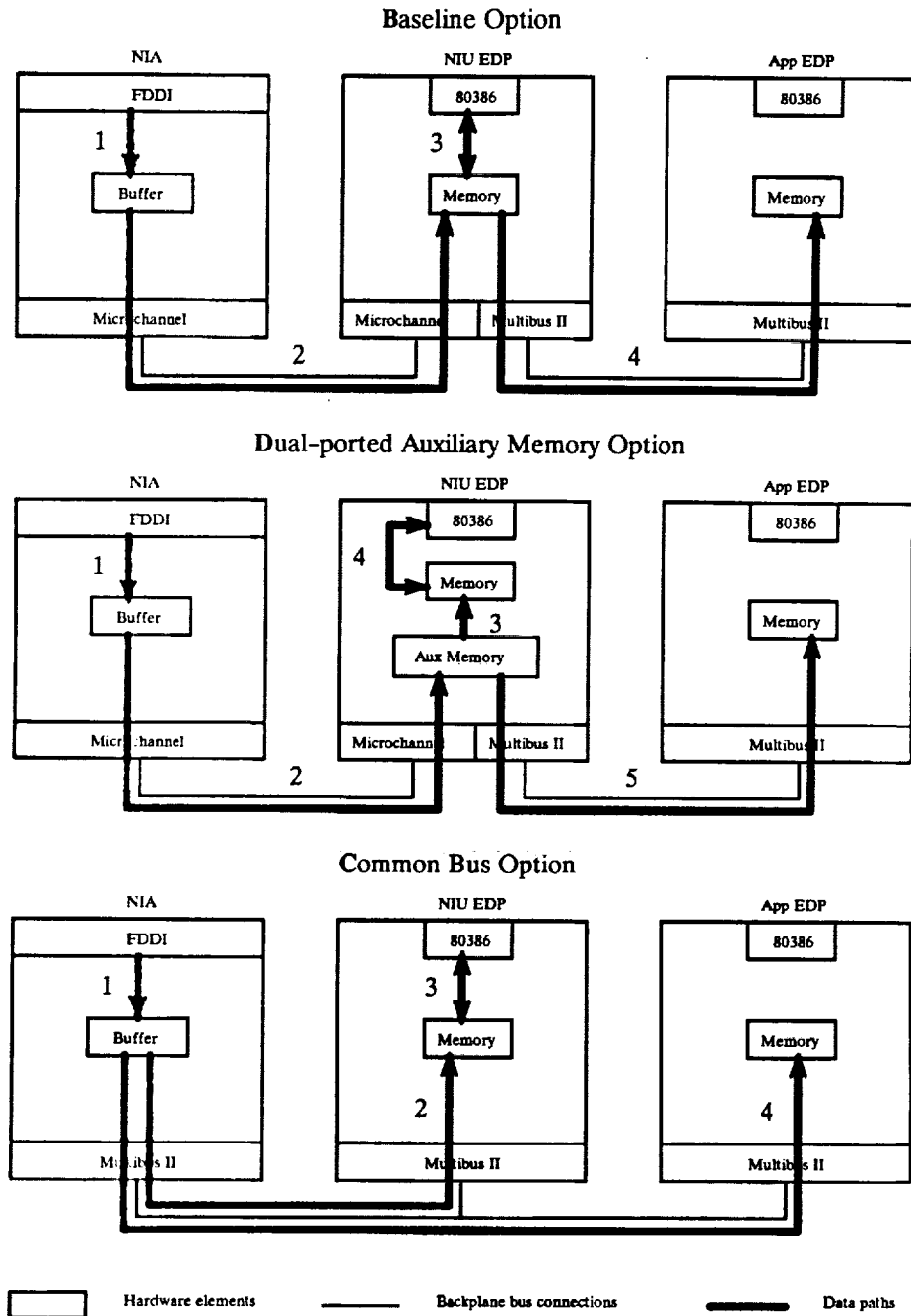


Figure 1
NIU Architecture Options

3 ANALYSIS

The purpose of the NIU subsystem is to move messages or *packets* from the FDDI fiber to the App EDP and vice versa. The subsequent analysis of the NIU is made relative to the packets it processes. In the DMS, two types of packets will be transmitted over the

network. The first type is an International Standards Organization (ISO) packet and the second type is a Consultative Committee for Space Data Systems (CCSDS) telemetry packet. In this analysis, a traffic mix which consists of 100% ISO seven-layer packets is assumed. Several further assumptions about ISO processing are necessary. First, no segmentation or reassembly of these packets is performed in the network, transport, or session layers. Second, transport checksums are not enabled. Finally, presentation layer encoding or decoding of packets is not performed in the NIU. These assumptions are consistent with current design of the WP-2 contractor for onboard, intra-DMS communications.

Packets themselves can be broken into two parts. The first part is termed the *header* and is used to maintain information necessary for the NOS to accomplish its function of effecting communication between applications on different processing nodes. Information contained in the header is formatted in terms of International Standards Organization (ISO) communication protocols and is used to perform such functions as packet routing and assurance of transmission reliability. The remainder of the packet is termed the *information field* and is reserved for use by the applications which are communicating with one another.

In analyzing the performance of these hardware architectures, two measures are important. The first measure is *latency* or the duration of time it takes a packet to traverse the NIU, in this case, beginning on the FDDI fiber and ending in the App EDP memory. Latency is measured in milliseconds. The second performance parameter of the NIU is *throughput* or the amount of data that the NIU can process over the same path specified for latency measurements during a given duration of time. Throughput can be described using two types of units, packets per second and bits per second.

Latency measurements are made relative to an individual packet which is passing through the NIU. This measure is important for individual applications which might rely upon a packet traversing the NIU in a duration which is less than some maximum value. Throughput measurements are made relative to the NIU system. This parameter indicates the overall performance of the NIU and is useful in providing information on how many applications may make use of the communications path during a specified duration.

In this paper, both of these measures are determined for the three architectures discussed in the previous section. The analysis which has been performed can be termed a *static* analysis. Equations have been derived which are based on an empirical evaluation of the architecture models. This approach yields best case numbers for latency and throughput measurements. A *dynamic simulation* of these system designs would yield measurements which would be somewhat less optimistic than these predictions. This is due to the fact that such an analysis would take into account contention for shared

resources which is not factored into this method. Section 3.1 discusses latency and section 3.2 describes the throughput analysis.

3.1 LATENCY

The latency of a packet passing through the NIU can be given as the sum of the durations of each of the path segments it must traverse within the NIU system. These path segments are different for each of the NIU architecture options.

In the subsections, 3.1.1 through 3.1.3, latency of a packet of data which is passing from the FDDI fiber to the App EDP memory is derived for each of the architecture options. In subsection 3.1.4, values for these latencies are calculated for varying packet sizes and any divergence between results is described.

3.1.1 Baseline Option

For the baseline option, this latency is made up of the following elements. A packet is first received by the FDDI chip set and transferred to the NIA buffer memory. This path is labeled **1** in the baseline option of Figure 1 and B_1 denotes its latency in the equation shown below. The entire packet is then transferred over the second path from the NIA buffer memory via the Microchannel bus to the NIU EDP memory. This path is labeled **2** and B_2 denotes its latency. The third path represents accesses by the 80386 processor to the main memory on the NIU EDP for both NOS program code and packet header data in order to effect communications processing. This path is labeled **3** and B_3 denotes its latency. Finally, the path which is labeled **4** and has its latency denoted by B_4 defines transfers of the packet information field from the NIU EDP memory to the App EDP memory which occurs after NOS processing has been completed. The entire latency of a packet traversing the path implemented by this approach can then be calculated as the sum of the latencies of each of these data path segments and is given by,

$$Lat_B = B_1 + B_2 + B_3 + B_4 \cdot \quad (1)$$

3.1.2 Dual-ported Auxiliary Memory Option

The latency of the second option which adds a dual-ported bank of auxiliary memory to the address space of the NIU EDP can be calculated in a similar manner. The latency of the path segment labeled **1** in the dual-ported auxiliary memory option of Figure 1 serves the same function as path 1 in the baseline option and will be denoted by D_1 . The path labeled **2** in this second architectural alternative is used to effect transfers from the NIA buffer memory to the NIU EDP auxiliary memory rather than the primary memory as in the Baseline approach. The latency for this segment is denoted by D_2 . Processing of header data which resides in the NIU EDP main memory can be accomplished at a more

rapid rate than data which resides in the auxiliary memory and as such, NOS processing is accomplished on the headers of the incoming packets only after the headers have been transferred to the NIU EDP main memory. This transfer is labeled **3** and its latency is denoted by D_3 . Once the transfer of the header is complete, the parameter D_4 is used to denote the duration required to effect NOS processing in the NIU EDP main memory which is labeled as path **4**. Finally, when NOS processing is completed, the packet information field is transferred from the NIU EDP auxiliary memory via the Multibus II to the App EDP memory. This transfer is indicated by path **5** and its latency is denoted by the parameter, D_5 . As with the baseline option, the overall latency of a packet traversing the dual-ported auxiliary memory option NIU can then be calculated by the following summation,

$$Lat_D = D_1 + D_2 + D_3 + D_4 + D_5 \quad (2)$$

3.1.3 Common Bus Option

The overall latency of this design can be derived similarly. As with the first two options, path **1** of the common bus option in Figure 1 represents transfers between the FDDI chip set and the NIA buffer memory. The latency of this transfer will be denoted by C_1 . A major difference in this design approach becomes apparent in paths **2** and **4**. In the first two options, an entire packet must first be transferred to the NIU EDP where it is processed by the NOS before the information can be transferred to the App EDP. In the common bus approach, path **2** effects transfers from the NIA buffer memory to the NIU EDP main memory just as in the baseline option, however, only the packet header is transferred. Path **4** represents the path the information field takes from the NIA buffer memory to the App EDP memory. Note that the information field is never copied to the NIU EDP memory saving the time necessary to effect that transfer. The common bus between the three cards make this approach feasible. In this option, C_2 will be used to denote the latency of the transfer of a packet header from the NIA buffer memory to the NIU EDP main memory and C_4 will denote the latency of the transfer of the packet information field from the NIA buffer memory to the App EDP memory. The remaining path, **3**, serves the same purpose as the equivalent path in the Baseline option. C_3 will denote the time necessary for the NOS to process a packet header as it resides in the NIU EDP main memory. As with the prior two options, total latency of the common bus approach is then derived as the sum of these latencies or,

$$Lat_C = C_1 + C_2 + C_3 + C_4 \quad (3)$$

3.1.4 Calculations and Results

To provide meaningful calculations of these performance measures, it was necessary to acquire numerical values which could be used for each of the individual data path

segments discussed in the above derivations. For this purpose, Summary Presentations for the Data Management System Preliminary Design Review (DMS PDR2) [1] which were provided by the WP-2 contractor, IBM, were utilized.

For those transfers which are made from the FDDI fiber to the NIA buffer memory, it was assumed that transfers by the NIA hardware can be accomplished at 100 Mbits/sec. Transfers across the Microchannel bus have been assumed to be accomplished at a rate of 2 octets/625 nsec. This rate is also used for transfers between the NIU EDP auxiliary and main banks of memory in the Dual-ported Auxiliary Memory option. Such transfers are assumed to be made in Microchannel 3rd party mode. Transfers across the Multibus II have been assumed to be accomplished using bus master mode at a rate of 4 octets/300 nsec. Finally, the duration of time the NOS requires to process a packet header is based on the provided instruction counts and 80386 processing rates. This Figure was calculated as 4.322 msec/header. The following table summarizes the basic parameters used to perform calculations for the previously derived equations,

| <u>Path Segment</u> | <u>Processing Rate</u> |
|------------------------------|---------------------------------------------------|
| Baseline | |
| 1 | 100 Mbits/second |
| 2 | 2 octets/625 nanoseconds (Microchannel 3rd party) |
| 3 | 0.004322 seconds/header (NOS processing) |
| 4 | 4 octets/300 nanoseconds (Multibus II) |
| Dual-ported Auxiliary Memory | |
| 1 | 100 Mbits/second |
| 2 | 2 octets/625 nanoseconds (Microchannel 3rd party) |
| 3 | 2 octets/625 nanoseconds (Microchannel 3rd party) |
| 4 | 0.004322 seconds/header (NOS processing) |
| 5 | 4 octets/300 nanoseconds (Multibus II) |
| Common Bus | |
| 1 | 100 Mbits/second |
| 2 | 4 octets/300 nanoseconds (Multibus II) |
| 3 | 0.004322 seconds/header (NOS processing) |
| 4 | 4 octets/300 nanoseconds (Multibus II) |

In order to perform the calculations for the various path segments it is important to observe that these latencies must be measured with respect to the amount of data that is operated on by the path segment. For instance, in the first two options, transfers between the NIA buffer memory and NIU EDP memory are required for the entire packet which may have a size which varies from 70 to 4500 octets. In these cases the latency of the paths is calculated as the product of the *total* packet size and the transfer rate of the Microchannel bus. In contrast, the common bus approach requires only the header be transferred from the NIA buffer memory to the NIU EDP memory. The latency of this

path then is given by the product of the packet *header* size, which is assumed to be 70 octets, and the Multibus II transfer rate.

It is important to note that for this analysis, packet header size is assumed to be constant despite the fact that actual packet headers will exhibit some variations in size. This assumption is justified by observing that these variations will be relatively small, in the range of 20 octets, and deviations introduced by such a small variance will have little effect on transfer latencies.

For the case of the NOS processing path segment, the time required to process a packet header also will vary according to the contents of the header. However, it is again assumed that these variations are slight with respect to the overall processing duration and therefore a constant value has been used.

The analysis was performed on the NASA/JSC ECF Cray Y-MP. The source code was written in standard C and is available. The specific equations used to calculate overall latencies for the given architectural options are listed in the source. The results of these calculations are represented graphically in Figure 2.

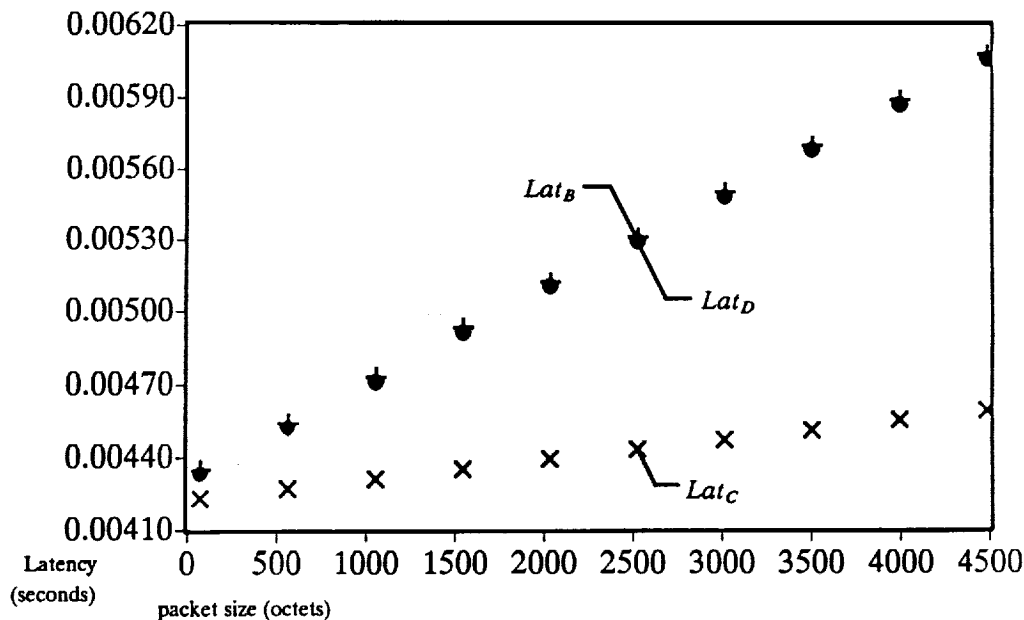


Figure 2
NIU Packet Latencies

It is interesting to note that in all three cases, NOS processing, which was assumed to consume approximately 0.0043 seconds/packet, overwhelmingly dominates the latency of the packet as it passes through the NIU. However, it is also worth noting the increase

in latency in the first two options of up to 25% as the packet size increases. This is directly attributable to the additional data copy necessary in these two approaches of the information field of a given packet from the NIA buffer memory to the NIU EDP memory. Furthermore, this additional copy is a direct result of the split bus design in the first two approaches. Only when the common bus approach is utilized can this additional copy be eliminated.

3.2 THROUGHPUT

At first glance, a simple reciprocal calculation on the latency Figures determined in the previous section might be performed to obtain a throughput Figure for each of the NIU architecture options. However, upon closer inspection, it can be observed that these systems are multiprocessing their functions. Since more than one task is being performed at a given instant, the system is actually capable of producing more throughput than would be indicated by analysis using the simple approach.

In order to determine the throughput of these pipelined, multiprocessing systems, it is necessary to determine the specific processing element which represents the bottleneck in the system that is, which segment of each pipeline represents the longest latency in the overall path. Throughput can then be measured as the reciprocal of the latency represented by the bottleneck. It is important to note that this approach represents the absolute best case throughput which can be obtained. For reasons which follow, it is very likely that worst case throughputs will be lower.

This analysis considers *shared resources* which are operated on by multiple processing elements. Therefore, accesses to shared resources must be *serialized*. For example, one processing element cannot be writing a value in a shared memory location while another processing element is reading that same location. The write and read must take place in serial order. By identifying the shared resources in the NIU system and analyzing the latencies incurred by any serialized accesses to those resources, the bottleneck can be determined.

As previously mentioned, this approach will yield best possible throughput results. There is a possibility of *contention* for shared resources. For example, consider the case of the NIU EDP in the baseline option at an instant when the NIU EDP processor is accessing main memory in order to process a packet header and a transfer from the NIA buffer to the NIU EDP memory becomes possible at the same time a transfer from the NIU EDP memory to the App EDP memory becomes possible. In such an instance, there must be some mechanism for arbitrating which transfer will be allowed to proceed once the NIU EDP completes its processing. The packet which must wait will have an additional latency introduced for that transfer. This occurrence is referred to as contention for the shared resource, in this case, the NIU EDP memory. Contention will

have the effect of reducing overall throughput in the NIU system; however, analyses which take into account contention by queueing requests for resources add another level of complexity to the analysis and have not been considered in this work.

This approach will be applied to each of the three architecture options under discussion. Sections 3.2.1 through 3.2.3 will analyze the respective approaches for their bottlenecks and then derive a throughput expression. Section 3.2.4 will provide calculations for these performance measures in terms of both packet and raw bit stream rates passing through each NIU option.

3.2.1 Baseline Option

In the current design, there are two shared resources. The first is the NIA buffer memory which can be accessed by either the FDDI chip set, the NIA onboard processor or the Microchannel drivers responsible for performing DMA transfers to the NIU EDP memory. The second resource is the NIU EDP main memory. This resource can be accessed either by the Microchannel bus which is completing the just mentioned transfer from the NIA buffer memory, the Mutlibus II drivers which perform DMA transfers from the NIU EDP memory to the App EDP memory, or the 80386 processor which processes packet headers as they reside in the memory.

The accesses which must be serialized in the NIA buffer memory include only the FDDI transfers and the Microchannel DMA transfers to the NIU EDP. Accesses by the NIA processor are not considered because it only processes packets which are used to accomplish station management functions and these packets do not leave the NIA. Therefore, serialized accesses for a packets passing through the NIA card include paths 1 and 2 of the Baseline option in Figure 1.

In the case of the NIU EDP, it is the main memory which represents the shared resource. Typical accesses to this resource were described at the beginning of this section and based on that explanation, serialized access for this card can be listed as paths 2, 3, and 4.

As noted in the previous section describing latencies, NOS processing of packet headers in the NIU EDP main memory consumes a duration which is much greater than any of the transfers which are required. This fact implies that the shared resource which is involved in the NOS processing, in this case the NIU EDP main memory, will represent

the bottleneck for any throughput measurement. With this in mind, the throughput for the baseline option can be expressed as follows,

$$\begin{aligned} Thrpt_B &= \frac{1}{\max(B_1 + B_2, B_2 + B_3 + B_4)} \\ &= \frac{1}{B_2 + B_3 + B_4} \end{aligned} \quad (4)$$

3.2.2 Dual-ported Auxiliary Memory Option

The throughput for the auxiliary memory option is derived in a similar fashion. In fact, the accesses to the first shared resource in this architecture, the NIA buffer memory are identical to that of the baseline option. The serialized accesses for this resource are given by paths 1 and 2 of the second option in Figure 1.

This architectural approach seeks to improve throughput by introducing additional parallel computations to the NIU system. The auxiliary memory which is logically a part of the NIU EDP but which physically resides on the NIA, attempts to provide a location from which DMA transfers can be accomplished to and from the NIU EDP while the 80386 processor is processing packet headers in the main memory. This additional parallelism becomes apparent when analyses of the interactions with the main and auxiliary banks of memory are performed. Accesses to main memory include only transfers of packet headers from the auxiliary memory to the main memory and NOS processing of the packet headers. These interactions are described by the latencies for paths 3 and 4. Accesses to the auxiliary memory are effected by DMA transfers across the Microchannel backplane. These paths include transfers from NIA buffer memory to the NIU EDP auxiliary memory and from the auxiliary memory to the App EDP memory. An additional access which is required for transfers of the packet headers to the main memory must also be serialized for this resource. Subsequently, the accesses to the auxiliary memory resource can be given by the latencies for paths 2, 3, and 5.

The bottleneck in this option is again the NIU EDP main memory since this is the resource in which NOS processing of packet headers is performed. The difference in this approach is that other accesses to the main memory resource have been reduced. Total throughput for this option can be expressed as follows,

$$\begin{aligned} Thrpt_D &= \frac{1}{\max(D_1 + D_2, D_2 + D_3 + D_5, D_3 + D_4)} \\ &= \frac{1}{D_3 + D_4} \end{aligned} \quad (5)$$

3.2.3 Common Bus Option

In the common bus approach, reduction of accesses to the NIU EDP main memory is again the theme, however, rather than add memory to the NIU EDP address space, this approach holds the information field in the NIA buffer memory while packet headers are processed in the NIU EDP main memory. Only when the headers have been processed is the information field transferred to the App EDP memory. The use of the common bus allows the information to be transferred directly to the App EDP rather than having to pass through the NIU EDP.

The shared resources for this option are exactly the same as those for the baseline. The difference is how those resources are accessed. The NIA buffer memory represents the shared element in the NIA but now has accesses from the FDDI chip set and the Multibus for transfers of packet headers to the NIU EDP main memory and transfers of the packet information fields to the App EDP. The involved paths for this resource are 1, 2, and 4 in the Common bus configuration illustrated in Figure 1.

The remaining shared resource is the NIU EDP main memory. In this option, only NOS processing of packet headers and transfers of packet headers are necessary accesses for this resource. These paths are listed by data path segments 2 and 3.

As with the first two approaches, the bottleneck in this design is the NIU EDP main memory which is where NOS processing occurs. However, like the dual-ported auxiliary memory approach, additional accesses to this resource have been reduced with respect to the baseline. Throughput for the common bus option can be expressed as follows,

$$\begin{aligned} Thrpt_C &= \frac{1}{\max(C_1 + C_2 + C_4, C_2 + C_3)} \\ &= \frac{1}{C_2 + C_3} \end{aligned} \quad (6)$$

3.2.4 Calculations and Results

The measurements made using equations (4), (5), and (6) were also based on the DMS PDR2 information. As with the overall latency Figures generated in section 3.1.4, it is important that the latency of specific path segments be calculated with respect to the amount of data that is transferred. Throughput calculations were included in the program generated for the Cray and the results of these calculations are graphed in Figure 3 and Figure 4.

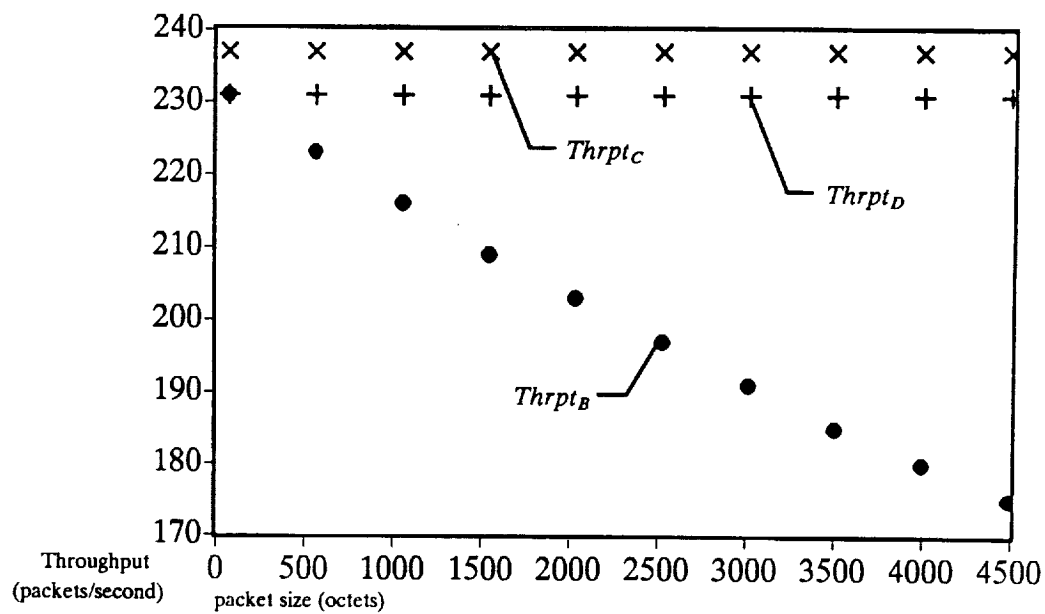


Figure 3
NIU Throughput (packets/second)

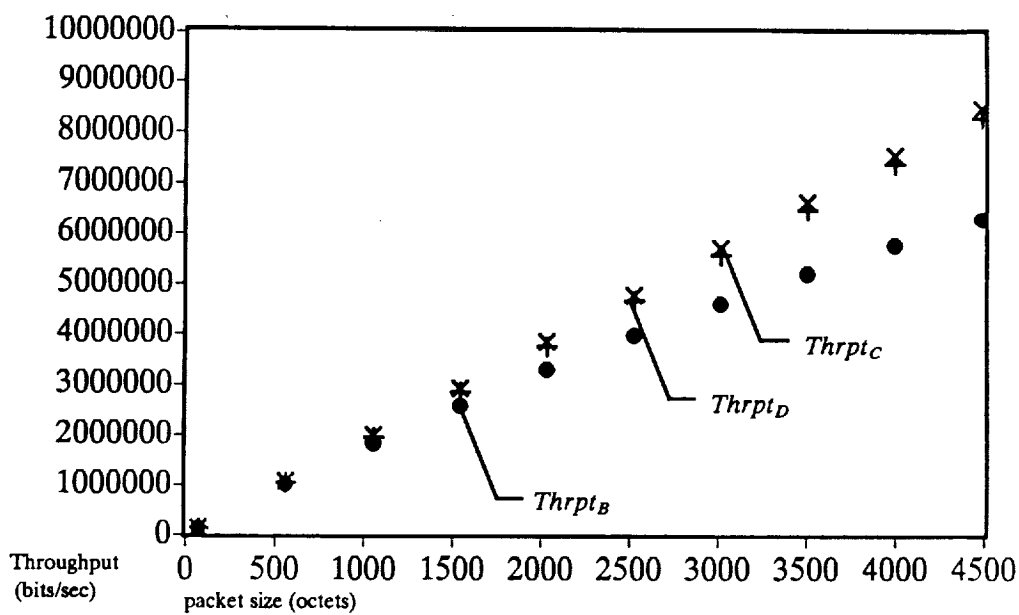


Figure 4
NIU Throughput (bits/sec)

It is apparent from these results that throughput for the common bus and dual-ported auxiliary memory options are similar. The small difference in the throughput results for these two approaches can be attributed to the different transfer rates used for calculations with the Microchannel and Multibus. The important result, however, is that both of these options show dramatic improvements in throughput over the current baseline design.

Another interesting result can be observed in these Figures. Note in the packets/second curves for the baseline approach that the throughput actually decreases as the packet size increases. The reason for this is the additional data copy from the NIA to the NIU EDP main memory. When this additional data copy is compounded with the overhead of NOS processing on the NIU EDP main memory as is the case in the Baseline option, this throughput result occurs.

The addition of the auxiliary memory which allows the additional data copy to be executed concurrently with NOS processing is IBM's approach at alleviating this result. However, when latency improvements are considered, the Common Bus design becomes more desirable.

4 IMPLEMENTATION IMPACTS

In addition to any measures of performance which might be obtained by a change in the design of the NIU system, it is important to consider what impacts a change in the design would have on implementation costs and schedules. It is clear that any change to the baseline design will incur penalties in both of these areas.

4.1 PHYSICAL IMPLEMENTATIONS

If either of the latter two options are to be adopted, there will be implementation changes needed in both hardware and software. It should be noted in the following descriptions that hardware changes are restricted to the NIA card in both design approaches.

To implement the dual-ported auxiliary memory approach, additional memory would need to be added to the NIA card along with circuitry to provide dual-porting of this memory. The common bus approach would likely require an increase in the size of the NIA buffer memory. As a consequence, since the FDDI hardware is capable of addressing only 256 Koctets, a paging mechanism would be necessary to allow it to address the buffer which would surely exceed that amount. The difference between the two approaches is that in the latter case, the memory would be added to the *address space of the NIA buffer* rather than the NIU EDP.

The common bus approach also requires a modification to the NIA backplane drivers. The current Microchannel design would be replaced by a Multibus II interface.

Such a change should require only a transplant to the NIA of the Multibus interface which is currently implemented on the EDP card.

Both of these design options require modifications to the NOS software. The use of auxiliary memory requires the NOS make use of the additional memory space. Management of DMA transfers from the NIA and App EDP which involved the NIU EDP main memory must make use of the auxiliary memory instead. Transfers of packet headers from the auxiliary to main memory must also be handled by the NOS.

In general, it can be said that the necessary modifications to the NOS for the common bus approach would be more significant than that of the dual-ported auxiliary memory approach. This would become particularly apparent if any of the underlying assumptions about the ISO network traffic were relaxed.

The necessary modifications can be described as requiring more complex pointer management within the NOS. Pointers are used to maintain the locations of the header and information fields of packets. Protocol functions are responsible for assembly and disassembly of packets using these pointers when necessary. Experience gained with the NIU prototype [4] has shown that this approach is not only feasible, even when the assumptions of this analysis are relaxed, but that the performance gains described herein can be realized.

Finally, the NIA firmware would also be impacted by the common bus approach in order to handle the more complex dual-ported NIA buffer memory. Such changes would manage any contention for the NIA buffer memory by DMA transfers involving the NIU EDP and the FDDI medium by the interface hardware.

4.2 COMMUNICATIONS AND TRACKING (C&T) INTERFACE

The impacts of the common bus approach would be more significant than that of the dual-ported auxiliary memory approach. However, there is another potential advantage associated with the common bus option which might be realized in the C&T interface to the Baseband Signal Processor (BSP).

It is clear that Standard Data Processors (SDPs) and Multi-purpose Applications Consoles (MPACs) can make use of the NIU design in its two card form, the NIA and the NIU EDP. However, both cards may not be necessary at the C&T BSP. A simplification of the C&T interface which was proposed by the Communications and Tracking Division [6] would make use of a single card, high-speed NIU which could be derived from the common bus design.

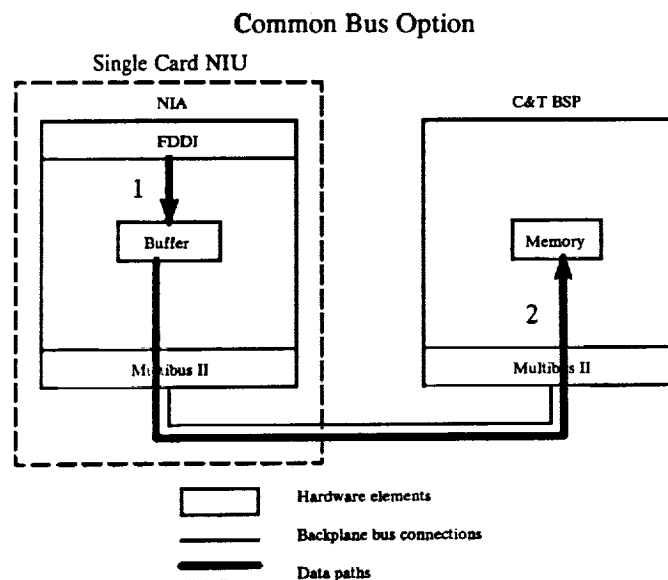
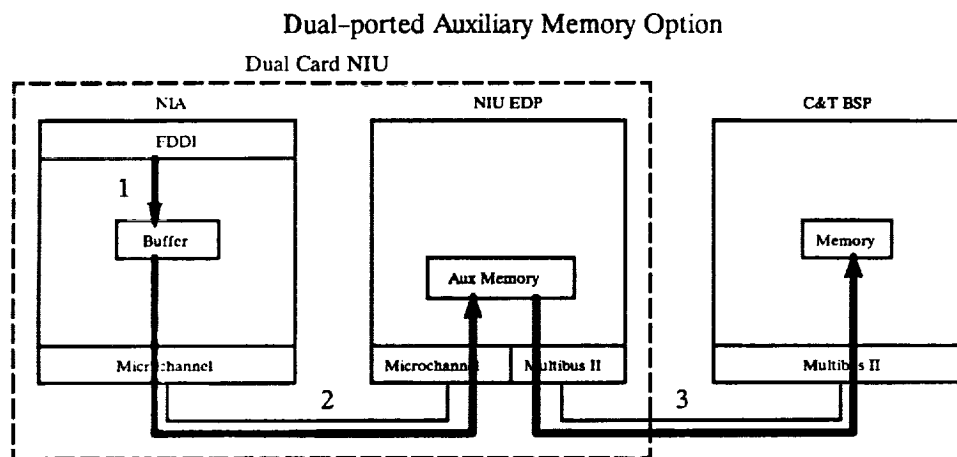
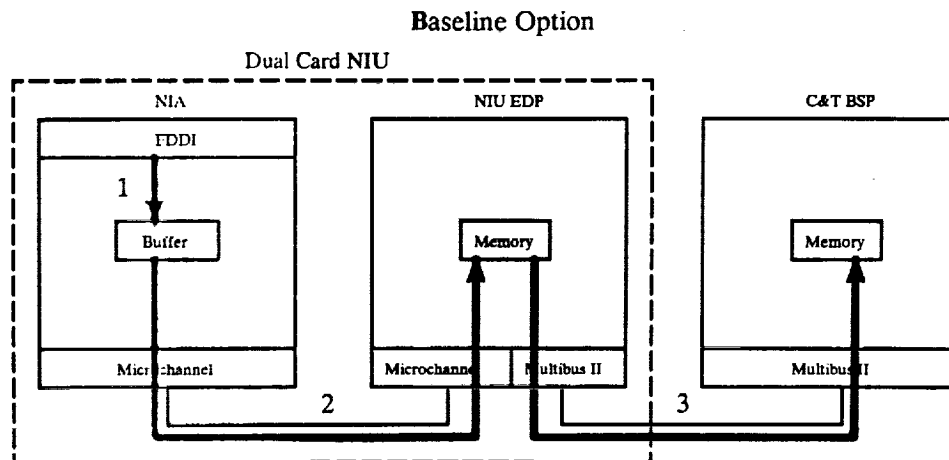


Figure 5
C&T Interface to the NIU Architecture Options

Data which passes through the C&T BSP will be of the type which was not considered in the preceding performance analysis, specifically, CCSDS telemetry data. Telemetry traffic uses only layers 1 (physical), 2 (data link), and in some cases 3 (network) of the ISO communications protocols to pass through the BSP and as such, it is not necessary to provide the entire NOS software for these packets.

There are, however, several major functions which need to be accounted for at this interface. First, some of the telemetry traffic will be ISO packets which are encapsulated in CCSDS packets. This traffic makes use of layer 3 to get to and from the network and must be encapsulated or deencapsulated in the BSP. Another function is the multiplexing and demultiplexing of variable length CCSDS packets to and from CCSDS transfer frames, respectively. Finally, there is the need to sort the CCSDS packets by virtual channel. These functions, except for virtual channel sorting, have been provided for in the alternate design proposed by C&T.

It should be clear that if the C&T alternate design were adopted and virtual channel sorting were moved to the ground systems, the inclusion of the NIU EDP at this interface would become superfluous. The only problem is that the current C&T BSP design specifies an interface to the NIU using the Multibus II backplane.

Figure 5 illustrates how the BSP interfaces to each of the NIU options. In each approach, the functions of the layers 1 and 2 protocols are implemented entirely on the NIA card. In the first two architecture approaches, telemetry data utilizes the NIU EDP to perform CCSDS functions before accessing layer 2 services which are implemented on the NIA. Use of a 4 million instruction per second (MIPS) machine to perform what may be considered relatively minor computational tasks is wasteful. The common bus approach which utilizes the simplified design proposed by C&T may eliminate the need for the NIU EDP in the BSP.

5 SUMMARY

The implications for performance and implementation have been presented. The dual-ported auxiliary memory design option presented by the WP-2 contractor, IBM, will increase throughput performance over the current baseline design. The common bus design option presented by Flight Data Systems Division will also provide the required additional throughput as well as a significant latency improvement over the dual-ported auxiliary memory approach. The common bus design also provides the additional advantage of allowing the adoption of the simplified design for the interface to the C&T BSP.

REFERENCES

- [1] Data Management System Preliminary Design Review No. 2, Summary Presentation Vol. 1, February 1990.
- [2] Firmware Requirements Specification (Network Interface Adapter), DRD SY-34.1, August 1990.
- [3] Interface Requirements Document (IRD) (Software), DRD SY-23.1, December 1989.
- [4] Multi-Compatible Network Interface Unit, Final Report, Honeywell Sensor and System Division, March 22, 1990.
- [5] Network Operating System Statement of Work (SOW), IBM, December 21, 1989.
- [6] Schadelbauer, S., "An Evaluation of the Space Station C&T/DMS LAN Interface", EE2-5/90-051, Tracking and Communications Division, NASA/Lyndon B. Johnson Space Center, May 31, 1990.
- [7] Software Requirements Specification (SRS) for the DMS Network Operating System, DRD SY-34.1I, May 1, 1990.
- [8] Space Station DMS NOS, Software Preliminary Design Document (SPDD), DRD SY-33.1, August 22, 1990.
- [9] Space Station Projects Description and Requirements Document, Volume 3: Projects Design and Development Requirements, JSC 31000, Draft C, October 19, 1989.



National Aeronautics and
Space Administration

REPORT DOCUMENTATION PAGE

| | | | | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|--|
| 1. Report No. TM 104735 | | 2. Government Accession No. | | 3. Recipient's Catalog No. | |
| 4. Title and Subtitle Network Interface Unit Design Options Performance Analysis | | | | 5. Report Date February 12, 1991 | |
| | | | | 6. Performing Organization Code EK4 | |
| 7. Author(s) Frank W. Miller | | | | 8. Performing Organization Report No. S-635 | |
| | | | | | |
| 9. Performing Organization Name and Address Flight Data Systems Division NASA/Johnson Space Center Houston, TX 77058 | | | | 10. Work Unit No. | |
| | | | | 11. Contract or Grant No. | |
| 12. Sponsoring Agency Name and Address Same | | | | 13. Type of Report and Period Covered Technical Memorandum | |
| | | | | 14. Sponsoring Agency Code | |
| 15. Supplementary Notes | | | | | |
| 16. Abstract <p>This paper presents an analysis of three design options for the Space Station Freedom (SSF) onboard Data Management System (DMS) Network Interface Unit (NIU). The NIU provides the interface from the Fiber Distributed Data Interface (FDDI) local area network (LAN) to the DMS processing elements. The FDDI LAN provides the primary means for command and control and low and medium rate telemetry data transfers on board the SSF. The results of this analysis provide the basis for the implementation of the NIU.</p> | | | | | |
| 17. Key Words (Suggested by Author(s)) Local Area Networks Network performance | | | 18. Distribution Statement Unclassified - Unlimited Subject Category - 62 | | |
| 19. Security Classification (of this report) Unclassified | | 20. Security Classification (of this page) Unclassified | | 21. No. of pages | |
| | | | | 22. Price | |

For sale by the National Technical Information Service, Springfield, VA 22161-2171

